REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Sept 95

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Architecture Reporting and Monitoring System (ARMS)

5. FUNDING NUMBERS

6. AUTHOR(S)
SMC/XRE
Lt John Kohler; Mr Bob Weber; Capt Scott Thomason

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SMC/XRE
Los Angeles AFB, CA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ AFSPC/XPA
ATTN: Maj Tim Gooley
150 Vandenberg St Ste 1105
Peterson AFB CO 80914-4650

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution is Unlimited

12b. DISTRIBUTION CODE

19970710 086

13. ABSTRACT *(Maximum 200 words)*

The ARMS software program purpose is to provide senior decision makers with "situational awareness" of national security space architecture by showing how future architecture alternatives fit within the current planning context for the joint military force structure. ARMS will demonstrate cost, schedule, performance, technical characteristics, and interdependencies o the national security space architecture using consistent terminology, methodologies, and display techniques. The concept of situational awareness that ARMS provides is achieved by having a complete picture of architecture cost, schedule, performance, technical characteristics, interdependencies through time, as well as, military worth, and cost effectiveness in an interactive graphical, personal computer-based environment.

14. SUBJECT TERMS
ARMS; Situational Awareness; Space Architecture Decision Support Tool

DTIC QUALITY INSPECTED 4

15. NUMBER OF PAGES
10

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Architecture Reporting and Monitoring System (ARMS)

Capt L. Scott Thomason
USAF Space and Missile Systems Center
Developmental Planning Directorate
2420 Vela Way, Suite 1467-A2
Los Angeles AFB, CA 90245-4659

Mr Robert Weber
The Aerospace Corporation
2350 E. El Segundo Blvd.
El Segundo, CA 90245-4691

Abstract

The need to improve military space architecture cost efficiencies has created the need to manage the space architecture as an integrated entity, and to examine tradeoffs in attributes of the architecture across not only traditional space mission functions (e.g., communications), but also across broader areas of responsibility (e.g., the entire space force structure). DoD decision makers therefore need an interactive support system to track the important cause-and-effect relationships among military utility, technical system performance, budget profiles, and schedules as they vary across a bounding set of future planning scenarios. This support system must also serve as an electronic communications medium among the diverse group of organizations that participate in the DoD planning process. The Architecture Reporting and Monitoring System (ARMS) is a customized integration of commercial off-the-shelf (COTS) software, currently under development at the Space and Missile Systems Center (SMC), that provides decision makers with the capability to "view" selected aspects of the space force graphically and navigate with real-time response through the multidimensional space architecture model. ARMS allows the user to rapidly assess gross impacts of major schedule slippage, program cancellation, launch failure, or transition to a new launch system. This paper describes ARMS, its development and capabilities.

Introduction

The Reinventing Air Force Space initiative (the joint reengineering venture of Air Force Space and Missile Systems Center, Air Force Space Command and Phillips Laboratory) and its follow-on, the Seven Strategies for Space, resulted in a new space architecture management structure. The new structure includes an overarching Space Architect, who is responsible for the people, processes and products for space systems, and two subordinate architects: the Space Mission Architect (SMA), who is responsible for mission design and execution, and the Space System Architect (SSA), who is responsible for system definition and integration.

The SSA provides comprehensive architecture management (physical architecture development, integration, and evaluation) to maximize efficiency of the modernization of space force structure over time. The SSA therefore must be able to comprehend the manifold interdependencies among cost, schedule, and technical performance across the entire military space architecture. ARMS was created to help satisfy the information needs of senior space system decision makers. The need for the capabilities that ARMS provides has always existed; only now is the technology mature enough to develop and implement the capability.

ARMS is the latest step in a long line of tools and processes aimed at development and management of the space architecture by SMC. The integrated requirements process (IRP) (and all the analyses and studies to support it) is crucial to the operation of ARMS; the integrated product development environment (IPDE) sought to capture the auto- and cross-correlations of systems from a "system of systems" perspective. The Future Strategy to Task to Acquisition (FSTA) work describes the methodologies essential to determining military utility. These activities serve not only as precedents to ARMS; some will provide continuous support.

Objective

The principal objective of ARMS is to provide to senior decision makers a form of situational awareness of the military space architecture and how future alternatives fit within the current planning context for the joint military force structure. The list of senior decision makers includes DoD space officials, operating command commanders, senior acquisition officials, and program managers. This list is not limited to space

officials; any decision maker who depends on space systems (in some form) is also included.

The concept of situational awareness is an extrapolation from traditional usage as the tactical commanders' integrated, real-time picture of friendly forces and their status; in this case the reference is to a fuzzier awareness of a much longer planning horizon of a decade or more, rather than a few hours or days. It is achieved by having a complete picture of architecture cost through time, architecture schedule, system interdependencies, and architecture performance. It also includes a measurement of return on investment, where decision makers can understand the effectiveness and military utility of the systems purchased compared to their cost.

In the Planning, Programming and Budgeting System (PPBS) process, the decision makers' need for information may not seem as immediate as in combat. However, the intense political pressures and limited attention allotted to each issue as more decisions are escalated to the highest level place a premium on providing senior leaders near real-time interaction with sources of credible data backed by line and staff organizations. Decision meetings of major stakeholders may last only a few hours, yet determine the fate of significant investments in future military capability.

As organizations remove management layers, there is a greater need for all participants, not just senior leaders, to have the same picture of the space architecture. Consequently, ARMS has a strong secondary objective of providing distributed, multiple access to data and data fusion capabilities as well as visibility into and control over the means of that fusion. ARMS seeks to bridge at least two major chasms among organizations in the acquisition process. One is the disconnect between the world of system engineering and military operations and the other, embedded in engineering management organizations, is the gap between scheduling specialists and budget planners. Bridging the cost/schedule gap, thereby tying cost to schedule, provides decision makers with a more immediate and understandable assessment of impacts caused by changes to either cost or schedule.

Development Approach

The ARMS Team has employed a "rapid prototype," or incremental, approach to development, in which team members develop a strawman list of perceived user needs, implement them in a concept prototype, then present the prototype to likely users (or their representatives). This allows for a short and frequent feedback loop with the potential user community. The results to date have been very positive; all of the potential users briefed so far have recognized the need for the capability, and have provided the insight necessary to improve the requirements list. As the tool becomes more robust, better-defined incremental builds will implement specific additional capabilities.

Based on the feedback received during demonstrations, potential users have stated that ARMS must provide the information necessary to perform a number of functions. First, ARMS can help support the concept of an integrated program objective memorandum (IPOM) for space systems by providing, in one place, the necessary cost and schedule information for the space architecture. Second, it can be used to compare alternative architecture concepts with existing and planned architectures, as well as how different system concepts fit into existing and planned architectures. Third, ARMS allows for visualization of various cost/utility trades, especially in presenting the "return on investment." Finally, ARMS provides for the visualization of how technologies support the end user needs. Each of these needs depends on a comprehensive set of space system information, aggregated into an architecture-level picture.

The system incorporates an open architecture, allowing for easy system expansion to accommodate new or modified system requirements. ARMS is being developed as Government-owned software using widely available, moderately transportable COTS products that are typically licensed by the Government.

Description

Overview

ARMS is an automated information integration and visualization system. It is a computer-based tool that takes individual space system data at its root level (e.g., system cost through time, system performance capabilities, system schedule, system technical characteristics, system interdependencies with other systems, etc.) and fuses it with information from rigorous off-line system and mission analyses. This fused information set is then presented to the user as an integrated, architecture-level graphical information display. Information from systems encompasses the entire system life cycle, from concept exploration through system retirement, while analysis results may apply only to a very narrow time-slice. ARMS is, in essence, an executive decision support system,

analogous to an information "Heads Up Display" (HUD) for space decision makers.

ARMS is not, by itself, an analysis tool. Rather, it relies on rigorous off-line analyses of systems and missions that are then aggregated into an architecture-level picture. The off-line analyses are performed with tools (such as GAP, STARFLEET, etc.) that are not part of ARMS. The data owners validate the analyses, which then become part of the system baseline description. ARMS then imports the results in a controlled, baselined way from the data owners.

Figure 1 shows a schematic block diagram of ARMS. The information fusion occurs in the top block (under Compare and Assess); all the blocks are essential to presenting an accurate picture of the architecture.

How ARMS works

To present architecture cost information, ARMS sums annual budgeted program costs into total annual architecture cost. ARMS presents architecture cost information in a consistent format, allowing the user to make "apples-to-apples" cost comparisons. The user may select which specific information to display (cost by system, budget category [color of money], segment, or mission area).

To present the architecture schedule, ARMS shows all individual program schedules in a single window using a consistent display format. The schedule display format includes program phases, major program milestones, and launch dates. The presentation format allows the user to immediately visualize the comprehensive architecture schedule.

To present an architecture performance assessment, ARMS relies on the results of external combat simulations and system performance models. The value added by ARMS is strictly in graphic query and display of stored data. The most desirable data to assess military utility would be provided by campaign level combat simulation results. However, lacking that, ARMS can function as an electronic form of the Requirements Correlation Matrix for each system in the database. ARMS facilitates aggregation of data from individual systems into a performance summary for each alternative architecture as a function of mission area, year, and conflict scenario. The summary utility assessment is presented as a "stop light" chart. The determination of whether any of the performance measurements is red, yellow, or green is accomplished by a group of stakeholder representatives using any mutually agreed process that could include such supplemental tools as Expert Choice or Equity.

Unique Characteristics

The uniqueness of ARMS consists more in the critical combination of characteristics than in any individual aspect. Other earlier software developments as part of the SAMS and FSTA projects addressed two or three of these attributes, from which lessons have been borrowed. However, until the recent availability of 90 MHz or greater CPUs and the integrated software suites which allow custom programming in an embedded, powerful, higher order language, it was not possible to achieve the desired real-time, interactive capabilities on a PC. The need for ARMS to both depend on and to support a distributed community necessitates running on a common, high performance, affordable, and portable platform. Experience with previous developments similar to ARMS indicates that the characteristics described here constitute a critical set. Accordingly, the ARMS concept prototype was developed and demonstrated using a 66 MHz i486 PC running Windows, Microsoft Excel, and Microsoft Project.

Real-time interaction & groupware function

The first and most unique characteristic of ARMS is that it supports a highly graphic user control and data display interface that executes data queries interactively. The research of Schneiderman at the University of Maryland has shown that, "Some innovations restructure the way people think and work....Experience with dynamic-query interfaces suggests that they are dramatically different from existing database query methods."[1] However, this leading-edge human-computer interface (HCI) work is based on custom code running on a UNIX work station. Achieving comparable manipulation of data objects and 100 millisecond response time on a Windows PC running modified COTS software requires efficient design and considerable understanding of the embedded macro language. It is in this last layer of integrative processing where data is graphically formatted so that patterns and conclusions leap out at the user. At this point it is essential to match the computer to human cognitive capabilities. The potential of using a computer, trackball and LCD projector to "fly around" in data-space and build alternative system architectures in a working group setting affords teams of stakeholders the opportunity of sharing discoveries that have, until now, been made by analysts working alone.

Multiple scenario basis for military utility

To have credibility with multiple organizations, any evaluation of military utility must be referenced to conflict scenarios characterized by such major environmental attributes as weather, terrain, water navigability, port and air base facilities, and such threat parameters as enemy order of battle for ground, air, sea and space forces, along with basic timing, target priorities, and scale of attack. The simple bookkeeping and linking functions of a decision support system, together with a well-designed user interface, facilitate the use of multiple threat scenarios to prevent tunnel vision, one of the major causes of poor decisions. Based on the widely referenced research of Kahneman, Tversky and Herbert Simon, Russo & Schoemaker have distilled ten barriers to successful decision making. Two of them -- "Lack of Frame Control" and "Shooting from the Hip"[2] -- are addressed by ARMS' capability of grounding all evaluation of system utility in some scenario context. In working planning issues characterized by large uncertainties, the analyst must frame a problem broadly enough to prevent being "blindsided" by a major factor omitted from consideration. Highly respected advanced planning organizations such as that of Royal Dutch Shell use a set of bounding scenarios to explore the envelope of possibility-space[3]. Bounding scenarios are used in the evolving IRP, from which this tool draws its requirements and military utility context. In ARMS' application to space force architecture planning we address a mix of mid-term and far-term projections as well as a variation in geographic location and intensity of conflict. Near-term scenarios less than 5 years into the future are not useful for architecture planning, as the systems to be deployed in that period are already in the acquisition pipeline. However, decision makers may desire a "State of the Architecture" briefing, in which current and near-term scenarios are used.

Hierarchical zoom in and out

One essential feature of any decision support system is that it follow the hierarchical structure of the area of interest. Since the work of Miller in the 1930s it has been accepted that human cognitive ability is limited by short-term memory to dealing with five (plus or minus two) variables simultaneously. Consequently, it is useful to simplify any analytic problem into layers so the number of variables in focus at one time can be kept under seven. The natural layers in the ARMS world of discourse are the overall space force or "architecture," "missions," such as navigation or spacelift, and "systems." The latter form the basic building blocks within ARMS. Any further detail in cost or performance analysis must be accomplished outside ARMS and imported as ARMS-recognized Measure of Performance (MOP) values (measurable performance characteristics) consistent with an accompanying life cycle cost profile.

The function of ARMS that permits users to build their own alternative architectures from systems and missions must also track important dependencies such as Satellite Control Network (SCN) services, launch vehicle and launch range, and Comsat links for dissemination of mission data. The ARMS user interface is constructed to provide feedback on such interdependencies as well as what level of the hierarchy is being viewed and queried. The user capability to instantly change the hierarchical level is an important HCI factor. Nobelist Joshua Lederberg commented on the creative process in science: "You have to be able to fantasize in rather crude ways--but then be able to shift from one frame of reference to another....Then there's a skill at combinatorial arrangements that comes up over and over again....--one has to have the skill to do a systematic, fairly rapid first scanning of the possibilities, of a given territory."[4] Appropriately, the emphasis with ARMS is on agility, scope and accuracy (and associated uncertainties) rather than precision.

Flexible user interface and control

A corollary to the importance of facilitating rapid movement across hierarchical levels is the ability of ARMS to place a minimum of constraints on user navigation along other paths. The user can begin by selecting a scenario, or merely accept the default and begin instead by viewing the integrated schedule of a selected space architecture. If understanding the source of military utility assessment is foremost, then the user can "drill down" through a particular mission and time-slice of the "stop-light" mission performance assessment matrix to find the linkage between an Operational Task Measure of Effectiveness (MOE) and quantitative system MOPs.

Control and display flexibility is even more important when attempting to move beyond the point of user acceptance to user empowerment. In encouraging planners to "think outside the box" it is crucial that any supporting tool provide the maximum freedom to seek new perspectives, follow new sequences of relationships, and discover new patterns in the data. This kind of support for creative exploration is an important part of avoiding one of the major decision errors discussed previously--being blindsided by an unanticipated consequence.

Experience on a previous project has demonstrated that user acceptance of a decision support tool is highly dependent upon the tool being responsive to a very wide range of user commands in any possible sequence. Most users have a very low tolerance of queries or commands that do not have a useful effect, or worse, precipitate a "crash" of the system. The ARMS user interface is designed to be user-friendly and robust to accommodate the needs of a wide variety of users and user perspectives.

Multi-windowed views and comparisons

Because ARMS relies on the human brain to perform any sophisticated data processing in the form of pattern discovery or pattern matching, ARMS must assist the user in generating a number of windows showing the different architectures being compared or different parameters being correlated in time (e.g., cost and military utility). The user interface should lighten the burden of scaling and aligning axes of the different graphics so that comparisons can be made easily. The use of multiple windows places a premium on display area so making efficient use of space needed for the control panel is important. ARMS employs a scrollable control panel to accommodate this need.

Example ARMS scenarios

The following scenarios are intended to highlight the practical application of ARMS. Though the examples are space system specific, one can easily extrapolate to larger operational contexts.

Suppose a senior decision maker wants to determine the current and planned military space architecture cost, through time to 2020, by either system, mission area, color of money, or segment (space, launch or ground). This information would be presented as a bar chart of cost through time, with each bar representing the total architecture cost for one year. Each bar is further subdivided into either system, color of money, or mission area, depending on user selection. The user may also want to see the schedule of each system in the architecture, shown on the same chart at the same time. This schedule would show not only milestones, but program phase, launch date, and expected system lifetime. Figure 2 shows an approach to presenting the cost and schedule information presented above.

The user may also want to understand how the architecture meets the documented needs (as defined by deficiencies documented in the Mission Area Plans (MAPs), Operational Requirements Documents (ORDs), and Mission Need Statements (MNSs)) of Air Force Space Command through time. The capabilities of existing and planned systems are compared with the needs of a given scenario and time. (E.g., an infrared sensor platform will perform differently against threat systems in Korea than it will against the same systems in Iraq; also, the threat is expected to continue to mature through time, negating or minimizing the performance of existing systems.) The information would be presented as charts showing the relative capability of the architecture to meet the needs in quantitatively-defined graphic representations (see Figure 3). This type of chart relies on system-specific data at the root level to quantify capability and the time when the capability will become available; the data are then aggregated to an architecture-level picture of capability.

What makes ARMS unique compared to other fusion tools is its ability to present information visually in forms that allow the user to see patterns in the data that would not otherwise be apparent. This provides the user with a much more intuitive understanding of the comprehensive data set. Because the data needed to manage the space architecture are inherently multidimensional (with such variables as cost, performance and schedule each having their own dimensions of color of money, mission area, time, etc.), one- and two-dimensional tools such as text, tabular numerical listings, or single graphic windows are insufficient for grasping the complexities and interdependencies of the space architecture.

Another unique aspect is the ability to dynamically change variables and quickly determine the impacts of the change. For example, if two new satellite systems under development each depended on the same launch vehicle for orbit insertion, any schedule slippage in the launch vehicle program (or even a launch failure) would have direct impacts on the schedules of the two new systems. What may not be immediately apparent, however, is the impact *to the architecture* of not having the two new systems operational. Depending on the specific missions of the new systems, various other elements of the architecture could be dramatically affected. In this example, suppose one of the systems was a new geostationary satellite that relied on a navigation satellite signal for autonomous ephemeris maintenance in order to reduce the size of the ground crew. There is now a multi-system interdependency that may not be apparent using traditional methods. In addition to the dependency of the satellite systems on the launch vehicle, there is also a dependency of one satellite system on another. One could further suppose

that one (or both) systems rely on still another satellite or ground system to provide communication connectivity to the end user. When fully developed and implemented, ARMS will allow the Space System Architect and other decision makers to visualize these interdependencies and to make better-informed decisions.

Finally, ARMS will allow the SSA to perform future space architecture trades to determine the best mix of systems and capabilities that meet both the warfighter needs and the budgetary constraints. In this way, new architectural concepts could be analyzed against one another much as system concepts are presently analyzed. ARMS is a tool that will provide the Space System Architect a large subset of the *information* needed to define and integrate the military space architecture.

Process

The ARMS team polled most of the SPOs at SMC to determine two things: whether the information necessary for ARMS was maintained by the SPOs as part of their day-to-day operations, and if so, whether the information storage format was compatible with an automated data retrieval and translation methodology. The poll results were mixed; the data is generally available, but the formats (and data definitions) vary. No SPO currently has an automated cost/schedule linkage. Many SPOs maintain schedules manually, using graphic presentation tools. Some programs use cost engineering models that link system performance to cost, but there is not a standardized output format. These issues present several challenges to the ARMS development team. The first is data normalization, where the same information is defined the same way across programs. This issue will involve operators and users, as well as developers. Another is the standardization of formats for information storage, presentation, and automatic retrieval. Still other issues include data timeliness, data quality, and security (classified data and proprietary information).

Though certainly a challenge, a notional process description for information retrieval, fusion, and presentation can be developed. For a given "what if" exercise or periodic status presentation, the ARMS team will essentially "go and get" the data that ARMS needs to provide the architecture-level picture. The team will do that by essentially logging into the file servers where the data is stored by the data owners, and accessing the current, approved program baseline data in a directory to which the team has been granted access. The data is copied from its source into the ARMS data repository (essentially a database). This data gathering activity could be manual or automatic; an automated approach is more appealing because of the length of time expected to take to gather the data manually. When all the information necessary for a specific query is gathered, ARMS is run using the complete data set. Note that in most cases, the root data must be translated in some fashion (to be in the same format, or to make all costs be based on the same year, for example). This translation will take place at the time the data is gathered.

With the complete data set immediately available, an ARMS operator can then display, in real-time, architecture cost, schedule, and performance information to a decision maker. The operator can select specific mission areas, specific architectures (collections of systems), applied to a selected scenario, at some point in time. This allows the decision maker to ask additional "what if" or "show me" questions, without having to wait hours or days for a response.

Standardization of language

ARMS directly addresses the difficult, long-term process of establishing a widely accepted methodology for evaluating military utility of dissimilar systems and missions. ARMS also implements a standardized lexicon, resulting in improved communication across the operator and developer communities. This is important because currently, there is not a standard lexicon, which adds months to many processes as operators and developers struggle to come to a common understanding.

Because the goal of developing a widely accepted analytic model of joint warfare is still being pursued on several fronts, what can be done is to obtain broad consensus on a language for classifying and relating the various important elements which constitute a complete representation of the military utility problem. The major elements can be generically described as operational goals, objects, attributes and behaviors. The critical achievement is to establish a dependency of the measure of accomplishing an operational goal (MOE) on the concept of operations (behaviors/rules) and physical architecture of the force structure as represented by measures of system attributes (MOPs) and functional interdependencies (behaviors/rules). It is a common understanding of this linkage of the hierarchy of operational goals and concept of operations to system and architecture performance attributes that bridges the current gap between the military operators and the system developers.

ARMS applies the methods used in the IRP to build an interface to the operational community. This work is based on the widely published framework for structuring operational goals--the Strategy-to-Task method derived by Lt Gen Kent and others at RAND[5]. The connection to the system engineering world is placed on a quantitative basis by documenting explicit relationships which link MOEs for each Operational Objective or Task to system MOPs at the architecture level. Using an object-based paradigm, each architecture inherits the MOP values for attributes of its constituent systems. Where interactions among systems create a new system-of-systems attribute, there may be additional MOPs at the architecture level. To promote visual understanding of the multiple MOPs which support a single MOE, we adopt the RAND approach of grouping the MOPs according to a sequential flow of basic military functions from Survey to Assess, Command, Control, Transport, and Engage. We add a multi-level decomposition of the primary functions down to a layer at which MOPs for such functional attributes as coverage, accuracy, timeliness and capacity can be specified to the system architect.

Cost/schedule linkage

ARMS addresses the current disconnect between schedule management and budget programming within the system program offices (SPOs). In recent years, specialization of each of these disciplines has prevented a simple automation of the dependency of cost on schedule modifications. For advanced planning purposes, the ability to automate approximate cost impacts from schedule changes is useful in assessing alternative architectures as a function of time. ARMS automates the "head count" method used manually in many SPOs. To support reprogramming of major schedule activities such as a satellite launch, ARMS will compute the cost of slipped activities by multiplying the "head count" level-of-effort by the average labor rate. Additional approaches using standard program cost profiles and links to program cost engineering models are being evaluated as alternatives for determining cost/schedule interaction.

Two-way conduit of demands on and results from more detailed models

Because ARMS sits at the top of the information pyramid presenting the most digestible, highly aggregated results to senior decision makers, it must be fed with data from more detailed cost engineering, system performance, operational architecture and dynamic combat models. Whenever important new results are obtained from detailed models, they will be uploaded into ARMS. When seen in the aggregate, the results from these models could precipitate a new round of more focused analyses, ones intended to respond to the "big picture" issues shown. This capability of linking outputs from detailed models to more highly aggregated ones could ideally be constructed into an efficient set of variable resolution models. This concept has been investigated by Davis, Hillestadt, Bankes and others at RAND[6]. They have documented many hazards in attempting shortcuts by linking legacy models without being careful to align the functional relationships so that they are logically compatible.

Distributed database

All data used to make ARMS work is owned and maintained by the data owners. In the case of SMC, data owners are generally the program managers of local system program offices. Although the ARMS team has attempted to minimize the impact to the SPOs (in terms of additional actions and expenses they must take in order to support ARMS), data owners would benefit by having their program information available in a common format to support a standardized information presentation format.

ARMS relies on a client/server approach to obtain the data it needs. Because the data is maintained by the data owners, ARMS connects electronically to the server on which the data is stored. The ARMS team would only have access to program manager approved data; this access would be granted by password protected file directories, for example. The data would also be maintained in the format that the data owner uses; this helps to minimize the cost impact to the program offices. ARMS, then, connects to all the required servers and downloads the necessary files to a local ARMS data repository

The proper approach for interfacing ARMS to external data sources is for analyst users to determine the kind of queries and integrated information displays that are meaningful to senior decision makers. Then the pertinent variables are decomposed into the dominant components for further sensitivity analysis at the system level in higher resolution models. By pursuing further investigation of only those key variables into constituent parts, the broader modeling community may concentrate its effort on issues of greatest decision making consequence. By propagating specifications of further analysis from ARMS downward, the users will have some confidence that methodological integrity of forthcoming results will be maintained.

In the near term, however, it may be often necessary to use results from legacy data bases or models which do not present data in the proper form for uploading into ARMS. The goal of making ARMS initially acceptable to a broad range of stakeholders means that it must be "backward compatible" with the distributed, program-based nature of data collection and trade-off analysis accomplished with detailed models. In this case, whenever possible, extra work will have to be accomplished to rigorously transform data outputs from legacy applications into the ARMS schema.

Summary

ARMS is an essential tool to helping senior decision makers better manage an integrated space architecture. With all the cost, schedule, and technical information for the entire architecture available in one place, at one time, senior leaders will have a better basis from which to make decisions.

Acknowledgments

The authors would like to acknowledge the contributions of the ARMS team members. ARMS is directed by the Systems Engineering and Integration Branch of the Developmental Planning Directorate (HQ SMC/XREI), Los Angeles AFB, CA. The following ARMS team members each contributed significant, unique talents and expertise to the development of ARMS and the writing of this paper: Jim Taniguchi, USAF; Doug Holker, Randy Crawford, and David Rudolph, all of The Aerospace Corporation; and Sam Goldstein of Sam Goldstein, Inc.

Footnotes

1. B. Schneiderman, "Dynamic Queries for Visual Information Seeking," *IEEE Software*, Nov 1994, p. 70.

2. J. E. Russo & P. J. H. Schoemaker, *Decision Traps: Ten Barriers to Brilliant Decision-Making and How to Overcome Them*, Doubleday, New York, 1989.

3. P. Wack, Scenarios: Shooting the Rapids, *Harvard Business Review* 63(6), 1985, pp. 139-150.

4. H. F. Judson, *The Search for Solutions*, Johns Hopkins University Press, Baltimore, 1987, p. 228.

5. R. H. Anderson, et al, *Toward a Comprehensive RAND Environment for Computer Modeling, Simulation, and Analysis*, Undated.

6. P. K. Davis, et al, *Variable Resolution Modeling Conference Proceedings*, Washington, DC, 1992.

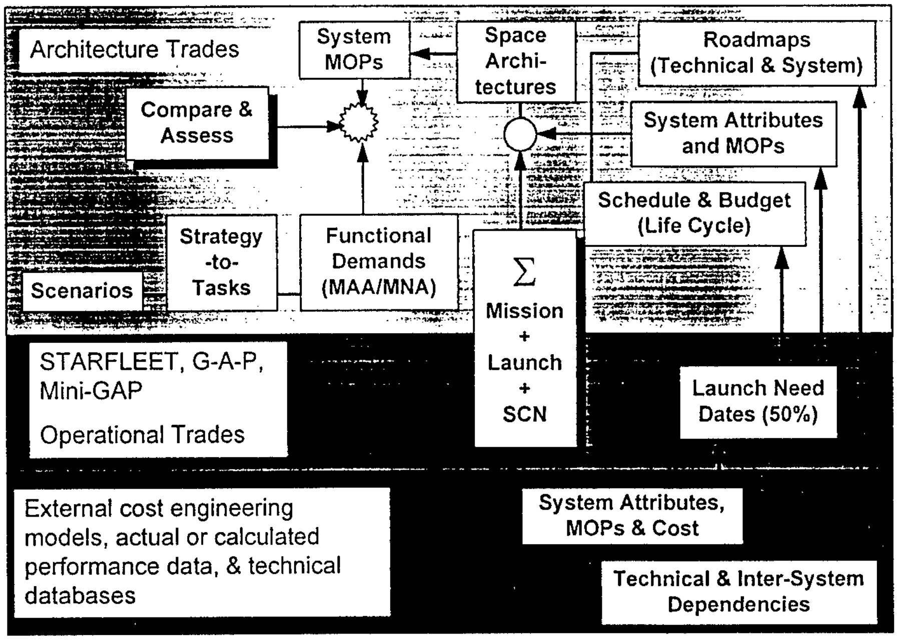

Figure 1 - ARMS Block Schematic

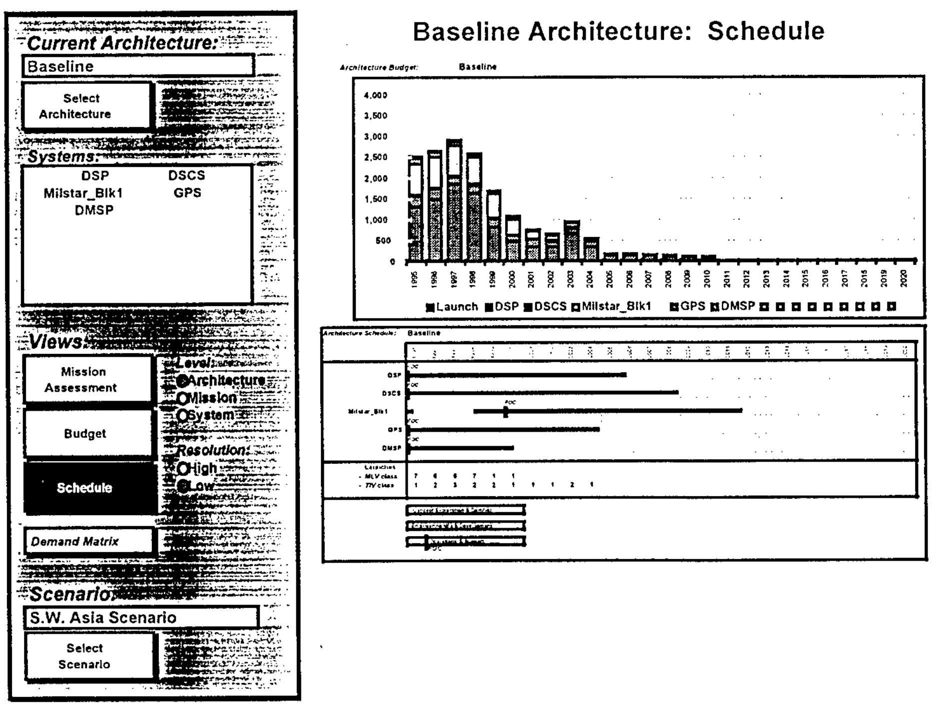

Figure 2 - Sample Cost and Schedule Windows

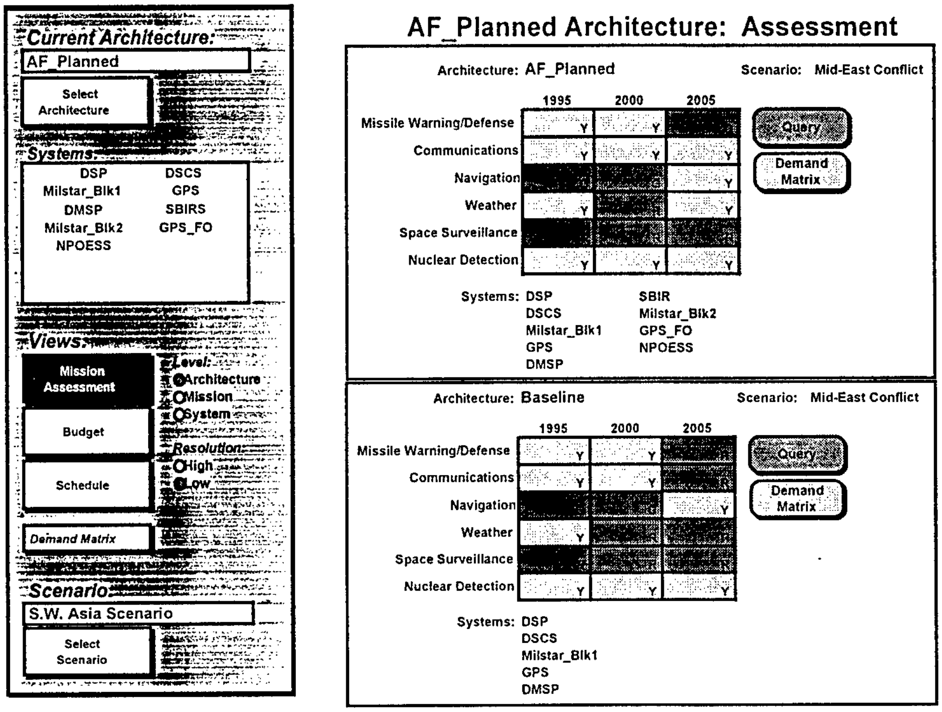

Figure 3 - Sample Architecture Assessment Windows